Evolving strategies for single-celled organisms in multi-nutrient environments.

Dominique Chu[1] and David J. Barnes[1]

[1]School of Computing, University of Kent, CT2 7NF, Canterbury, UK
dfc@kent.ac.uk

Abstract

When micro-organisms are in environments with multiple nutrients, they often preferentially utilise one first. A second is only utilised once the first is exhausted. Such a two-phase growth pattern is known as *diauxic growth*. Experimentally, this manifests itself through two distinct exponential growth phases separated by a lag phase of arrested growth. The duration of the lag phase can be quite substantial. From an evolutionary point of view the existence of a lag phase is somewhat puzzling because it implies a substantial loss of growth opportunity. Mutants with shorter lag phases would be prone to outcompete those with longer phases. Yet in nature, diauxic growth with lag phases appears to be a robust phenomenon. We introduce a model of the evolution of diauxic growth that captures the basic interactions regulating it in bacteria. We observe its evolution without a lag phase. We conclude that the lag phase is an adaptation that is only beneficial when fitness is averaged over a large number of environments.

Introduction

When bacteria (and other organisms) are presented with two (or more) nutrients of different kinds then they often show what is called, diauxic growth (Deutscher, 2008; Brückner and Titgemeyer, 2002). The discovery of this effect goes back to (Monod, 1949) and means that a cell takes up one nutrient first and exclusively. Only when this first nutrient has run out will it turn to the second. This first is usually preferred and affords higher growth. In wet-lab experiments diauxic growth is a well known, frequently encountered effect. In growth experiments it manifests itself through a signature fast initial exponential growth, followed by an episode of reduced or no growth, and then a second phase of exponential growth. The first growth phase is fuelled by the preferred nutrient; the second growth phase by the less preferred one. The phase of arrested growth in between is usually called the *lag phase* and can be very long compared to typical bacterial generation times. For example, in the case of the *E.coli* and glucose-lactose growth, the lag phase is of the order of a typical generation time (about 20 minutes).

Within biosciences, diauxic growth is a well known phenomenon. Its mechanisms are well understood and certainly its experimental phenomenology is routine for the working micro-biologist. Yet here we are interested in exploring the evolutionary origin of the effect. The standard account for diauxic growth is that it enables cells "to increase their fitness by optimizing growth rates in natural environments providing complex mixtures of nutrients" (Stülke and Hillen, 1999). This intuition is likely right. Ultimately, all it states is that diauxic growth helps to maximise growth, which, at least in the context of bacteria, equates to saying that it is a beneficial adaptation. However, what it does not state is how and in what way diauxic growth helps to maximise growth. What precisely are the conditions under which a two-phase growth strategy is better than simultaneous uptake of both nutrients?

One of the aspects that is particularly puzzling from an evolutionary point of view is the presence of the lag phase. Arrested growth during an exponential phase is costly, especially if it takes as long as a typical doubling time. It is not clear how this long phase of delay can be reconciled with the justification of diauxic growth as a strategy that maximises growth. Naively, one would assume that there are a number of strategies that could avoid the lag phase and still reap the benefits of diauxic growth. For example, a mixed strategy that relies on a small amount of uptake from the second nutrient while the first is still available. Or, a more sophisticated regulatory mechanism where uptake of the second nutrient is activated as the first nutrient runs out. This suggests that it is worthwhile to re-consider the evolutionary origin of diauxic growth.

Recently, there has also been some renewed experimental interest in diauxic growth. New methods in single-cell observation now make it possible to see what happens in diauxic growth at the level of the individual cell. There is now evidence (Boulineau et al., 2013) that the growth phases and the lag phase at the level of the individual cell are not as well defined as they are at the population level. For example, throughout the (population level) lag phase, some of the individual cells continue to grow without delay while others cease growth for substantial amounts of time before resuming growth well into the second growth phase. This suggests that diauxic growth is a population level effect, while at the

level of the individual the dynamics are determined by noise and heterogeneity of the population. In turn, this suggests that in order to understand diauxic growth and its evolution it is necessary to combine population level dynamics, stochastic modelling and artificial evolution. This will then make it possible to understand which strategies emerge under which assumption.

To this end, we created a program that combines biochemical simulation techniques with agent-based methods. It can simulate populations evolving cell populations. The model we created with the program simulates each individual cell using a discrete stochastic simulation algorithm. Cells in the simulation are exposed to external nutrient that they need to take up and metabolise to grow. Once they have grown to a sufficient size they will divide. Over time a population of agents emerges. We also experimented with two types of evolution: an explicit approach using a genetic algorithm, and an implicit fitness approach. In the implicit approach, we monitor multiple concurrently-evolving populations with limited interaction/exchange between them.

In our model we simulate a scenario where two nutrients are offered to the cells: N1, which is the high quality nutrient, and N2 which gives only half of the energy per molecule when compared to N1. For all our simulations we kept the structure of the internal cell dynamics fixed and only allowed the system to change regulation by adjusting parameter values. We find a hybrid strategy evolving: While there is a clear diauxic shift observable, we do not see an initial phase of growth fuelled solely by one nutrient. Instead, during the initial phase there is a certain amount of uptake and usage of both nutrients. However, the better quality nutrient, is taken up at a higher rate. Once the better nutrient is exhausted, the uptake rate of the second is increased strongly. Crucially, this happens without any delay—there is no lag phase evolving in our simulations and it depends crucially on the uptake/metabolism being capacity limited. This is in line with the intuition mentioned above: that a lag phase comes at a substantial fitness cost.

These results suggest that diauxic growth is a general phenomenon that will evolve as long as there is a limitation on the capacity of the uptake system. However, our results contrast with the strictly sequential nature of diauxic uptake in real bacteria. We hypothesise that this needs to be explained by life-time averaging of fitness.

Methods and Models

The Model of a cell

We used a minimal model of a cell that can take up two types of external nutrient: N1, N2. The internalised nutrients are E1, E2. They can be metabolised into internal energy E0 with rates lk1 and lk2 respectively. This energy is required in order to express the porins P1 and P2. These are required to take up N1 and N2 respectively. Furthermore, internal energy E0 is required in order to produce biomass and growth. Growth is essential because the cell is only allowed to divide if it has reached a minimum size. As such, this model captures the fundamental trade-off that real cells also face: How much energy should be invested into growth and how much into maintaining the machinery necessary to take up nutrient? If everything is invested into growth then the cell will soon end up in a position where it cannot replace its uptake machinery. On the other hand, if it invests all into the uptake machinery, but none into growth, then it will not be able to divide and hence will be evolutionarily unfit.

Here we are interested in the dynamics of diauxic growth. Hence, we built into our model an interaction mechanism between uptake of nutrient 1 and nutrient 2. We modelled this mechanism loosely on the PTS uptake system in bacteria (Boianelli et al., 2012). Nutrients 1 and 2 can only be taken up by porins specific to them. Uptake of the preferred nutrient leads to the dephosphorilisation of a repressor R^*. Its dephosphorylised form R can bind with porin P2, thus blocking uptake of N2. The porin-repressor dimer disintegrates with a rate of ukb, thus restoring the ability of the porin to take up nutrient again. Note that R^* is not explicitly modelled but assumed to exist in constant abundance.

Cells divide with a given rate of 1 once they have reached a certain size threshold. Upon division the protein and porin content of the cell is randomly (but typically not equally) divided between parent and offspring. In our model we have no explicit degradation of proteins or porins. Instead we rely on dilution through division as a mechanism to control particle numbers per cells. This choice is, on the one hand, realistic with respect to real bacteria but it also reduces the model complexity by saving two additional parameters.

Restrictions on uptake mechanisms In this contribution we considered two variants of the model. One where the cell is limited in its capacity to take up/metabolise nutrients. Here, we represent this by a limit on the number of porins that the cell can incorporate into its surface. This is represented as a negative feedback from the number of porins to rate of porin expression. In the formal description of our model in Table 1 this restriction is implemented by the second term of the rate in the expression of P1 and P2 (sixth and seventh reactions). Below, we will also report simulations where the restriction on the porin number is disabled. This is realised simply by removing the second term.

Evolution

We considered two types of evolution: implicit and explicit. For both we varied the following parameters from Table 1: lk1,lk2,K1,K2,leak1,leak2,g,KG,dR,kb,ukb. All parameters are allowed to vary in the arbitrarily chosen but constrained floating-point interval $[0, 15]$. Only the parameters dR, kb, ukb regulating internal signalling reactions are multiplied by 10 before being used. This reflects that they are are happening on a faster time-scale

Description	Substrate	Product	Rate
N1 taken up	`N1`	`E1 + R`	`[P1 hill(N1/volume,K,2)]`
N2 taken up	`N2`	`E2`	`[P2 hill(N2/volume,K,2)]`
E1 converted to energy	`E1`	`E0`	`[lk1]`
E2 converted to energy	`E2`	`E0`	`[lk2]`
E2 loss through inefficiency	`E2`	$\emptyset$	`[lk2]`
Porin 1 production	`E0`	`P1`	`[(leak1 + hill(E1,K1,2)) .` `(1-hill(P1+P2,surfaceArea/50,4))]`
Porin 2 production	`E0`	`P2`	`[(leak2 + hill(E2,K2,2)) .` `(0-hill(P1+P2,surfaceArea/50,4))]`
De-phosphoryilisation of repressor	`R`	$\emptyset$	`10 dR`
Repressor blocking porin 2	`R + P2`	`B`	`10 kb`
Decay of repressor-porin compound	`B`	`R + P2`	`10 ukb`
Volume increases by one unit (growth)	`5 E0`	`volume`	`g`
Division happens when volume reaches 50	`50 volume`	Division	`1`

Table 1: The definition of the model. The expression `hill(x,y,z)` abbreviates the function $x^z/(x^z + y^z)$. The last column gives the rate (when the expresssion is in square brackets) and the rate constant otherwise.

than gene-expression. The implicit scenario consisted of 32 environments. Each is populated by 5 random cells. A certain amount of N_1 and N_2 was given to each environment. Within each environment the cells then grew, took up nutrients, and divided when they reached the division threshold volume (set to 50). On division, cell volume and contents were randomly divided between the parent and offspring. The offspring cell underwent a mutation with a probability $P_m = 0.005$. Mutations are adjustments of the parameters by adding/subtracting up to 10 percent from the current value of the parameter (while respecting the constrained ranges.) Hence, over time the diversity of the population grows within each environment. Once all nutrient is exhausted the simulation of the environment is halted.

We simulated the environments in parallel and once all had exhausted their nutrient we performed a migration step. In all environments we first chose a small number of migrant cells, removed those from the population and then pruned the remaining population in each environment back to 5 cells by randomly discarding cells. Following this we chose random destinations for the migrant cells. Migrant cells replaced a random member of the existing seed population. The migration probability was set to 0.001 per iteration per cell. On average, this resulted in less than one migrant per environment with the settings we used. Once these steps were completed, each environment was simulated again and the process repeated for the desired number of generations.

Explicit evolution was implemented by using a standard GA with fitness proportional selection and a population size of 32 environments. The GA was run for 300 generations. Each solution in the population was represented by the parameter values of the model. Each environment was seeded by a single cell that was then grown on a mix of nutrients as explained above. Once a generation was completed, the final population size was recorded and used as the fitness for the corresponding solution.

In this article we report a competitive GA. This means that after a first GA evaluation of 300 generations, we restarted the evolutionary process. Yet, this time, we seeded each environment with a candidate solution (which evolves) *and* the best solution found during a previous evolutionary run. We only evaluate the fitness for the candidate solution. The fixed solution only serves as a competitor against which the evolving solutions have to persist. We performed several series of these competitive GAs whereby, in each iteration, the candidate solutions are evolved against the best solution emerging from the immediately preceding competition.

The Modelling Software

The simulation was performed using custom software developed in Java to support the general modelling and evolution of cellular species whose dynamics are defined by a parameterised set of reaction equations. This software combines agent-based modelling with stochastic simulations of biochemical networks. The user only needs to define the reaction network of one particular cell/organism and the number of instantiations of this cell. The software then automatically creates the relevant reaction networks to simulate each of the distinct cells.

A model is realised by an efficient implementation of the Gibson/Bruck stochastic simulation algorithm (SSA) (Gibson and Bruck, 1998) — a variant of Gillespie's algorithm (Gillespie, 1977). Since each cell in an environment potentially represents a separate parameterisation of the model's equations, and the population size will vary within a single generation, the number of reactions to be managed is dynamic and can grow very large. A pure implementation of Gillespie's SSA would recalculate the propen-
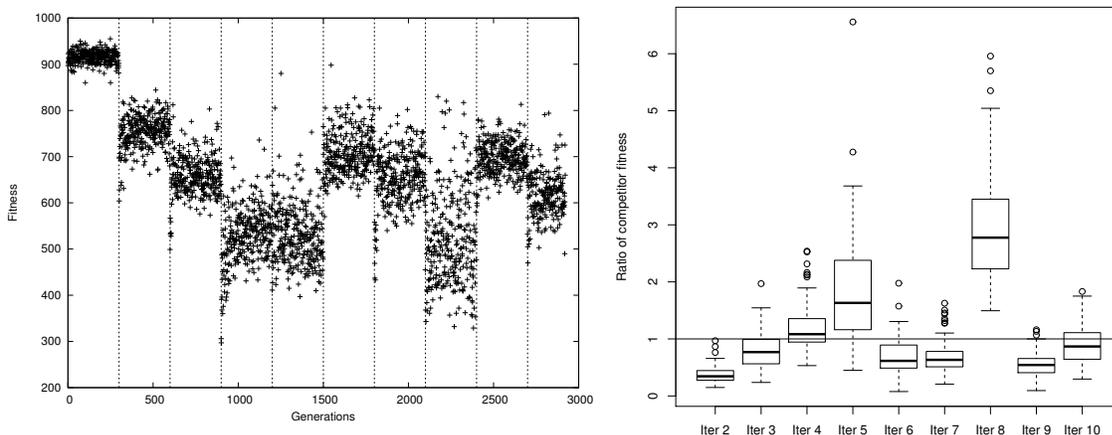
Figure 1: The evolution episodes in the GA. (**left**) The vertical bars indicate the fitness episodes. Initially, there is a fall in fitness. This is followed by a quasi oscillation of high and low fitnesses. (**right**) The ratio of the competitor fitnesses for 100 simulations of the best solution from the GAs. A value < 1 indicates that the evolved solution in the current iteration is better than the solution in the previous iteration. In the second iteration the solutions outcompete the incumbent from the previous iteration. In the third iteration this is also the case, in general. However, for higher iterations competitive solutions fail to evolve, leading to a quasi-oscillation of competitiveness.

sity of *every* reaction after the occurrence of a single reaction and this would be very inefficient. For instance, many reactions of an individual cell would have no impact on the propensity of the reactions in other cells; only those involving the shared external nutrient have an impact on all cells within an environment. Reaction interdependencies are determined from the reactions and rate expressions at the start of a simulation and only the propensities and next reaction times of dependent reactions are updated after a reaction takes place. In addition, a priority queue is used to order the occurrence of reactions, making identification of each next reaction a simple operation.

The software allows a model to be specified in the form of an arbitrary number of reaction equations. The reaction rate may be either a constant or an expression. The specification has separate sections for those elements that are internal species names (e.g. `E1`), evolvable parameter names (e.g. `lk1`), named constants or expressions (e.g. `surfaceArea`). Rate expressions may involve any of these elements. An arbitrary number of species may be designated as shared resources (such as `N1` and `N2`) along with distinctive nutrient 'values' (i.e. energy value to a cell). The evolution of a parameter's value may be either constrained or unconstrained within an arbitrary non-negative floating-point range.

At the start of a simulation, a model is read from a text file and a further efficient feature of the implementation is that the model is converted into a Java class definition that is dynamically compiled at runtime. Concurrency is supported in the software through both multi-threading and the ability to distribute environments across multiple networked machines.

A generation is completed within an environment when any one of the following conditions is fulfilled: all of the cells have died; a time limit has expired; all of the external nutrient has been consumed; no cell within the environment is capable of any further action (i.e., the total propensity of the reaction set is zero). Depending on the requirements of the simulation, when a cell divides there can be a probability of mutation of its parameter values for the daughter cell — as used in the implicit evolution scenario of our model. In addition, crossover is available when a cell is migrated into an environment with an existing population.

Results

For the purpose of this study we made no assumptions about the values of the parameters. Instead, all parameters were evolved using either the explicit or the implicit evolution. We found that the optimisation problem itself is not particularly challenging and the evolutionary algorithms found viable solutions quickly. Indeed, in the implicit evolution the fitness reached close to its maximum value within a few generations.

However, the raw fitness value reached is not necessarily an indicator of the competiveness of the results. A very high fitness solution may perform poorly when competing

Dominique Chu, David J. Barnes (2015) Evolving strategies for single-celled organisms in multi-nutrient environments. Proceedings of the European Conference on Artificial Life 2015, pp. 226-233

directly against another solution. This can be illustrated by the sample evolutionary run in Fig. 1 which shows the results of an iterative GA evolution. The first iteration of this simulation produces a solution that, by itself, performs well and reaches a high number of cells. However, when competing against the solution obtained from the second run it will only grow to about one third of its competitor's level. In this sense, the second solution outcompetes the first.
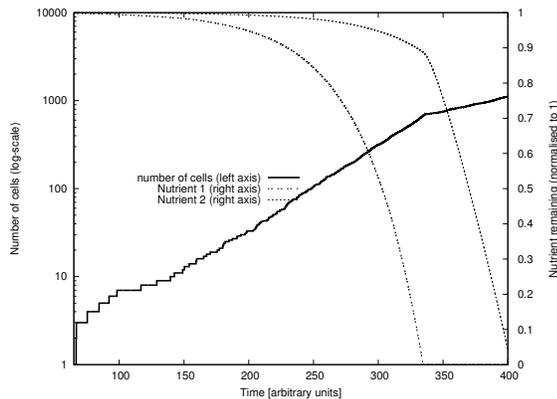

Figure 2: A simulation of a population. The growth of a population over time (left axis) shows a distinct two-phase growth pattern indicated by two straight lines in the logarithmic plot. However, note that there is no lag phase. Initially there is small simultaneous uptake of nutrient. Once nutrient 1 is exhausted the uptake rate of nutrient 2 is increased.

The difference between these two solutions is predominantly the speed with which the nutrient is taken up. This is apparent from the curves labelled "alone" in Fig. 3; these show the two solutions simulated on their own and demonstrate that the first solution grows over much longer time than the second. This is also confirmed by a closer analysis of the simulation data (not shown) and is a feature that is robustly reproduced by repetitive evolutions. Apart from the speed of the uptake, there is also a slight difference in the uptake priority. While the first evolved solution takes up N2 faster than N1, this order is reversed in the simulation of the second solution. Yet, in both solutions there is largely simultaneous uptake for most of the time.

More interesting is the change in behaviour from the second to the third iteration. In this step, there is not only an increase in the speed of uptake, but also a fundamental change in the way nutrient is taken up. In the first two solutions the repression of P2 by R was effectively disabled by a suitable parameter choice. On the other hand, in the third solution the parameters evolved so as to activate this repression mechanism. Specifically, this affected three parameters: the dephosphorylation rate dR; the association rate of the repressor and porin 2 kb; and the dissociation rate ukb. Effective repression can only happen when the dissociation rate is low compared with the association rate. Additionally,

repression requires the repressor R to be available. If dephosphorilation is too fast compared to the association rate kb then effective blocking of porin 2 will not be possible.

The parameters of the third solution evolved so that uptake of the N2 is strongly reduced while N1 is taken up. N2 is taken up at a much higher rate once N1 runs out. However, note that this is only a partial blocking. There is still a low level of simultaneous uptake of both nutrients (Fig. 2). In this particular series, solutions return to simultaneous uptake of nutrients after the third iteration. In different series of evolutionary runs one would observe a qualitatively similar behaviour, although precisely when the regulated diauxic growth appears will vary between runs. After the first regulated diauxic growth appear, subsequent evolutionary iterations lead to quasi-oscillations between very fit solutions and rather poor ones. Again, this qualitative feature robustly appears in repeated runs (data not shown).

The evolution with genetic algorithms illustrates well how diauxic growth evolves. This can be summarised in three steps: At first, solutions evolve to maximise utilisation of nutrient, irrespective of the relative value of nutrients. Then, in a competitive dynamics, solutions must become faster to prevent competitors from taking up nutrients. At this stage uptake may still be simultaneous. Finally, in a third step solutions evolve sequential uptake of nutrient. This third step depends crucially on the space on the surface of the cells being limited. If there is unlimited space, then only the first two stages will appear and no diauxic growth evolves. This stepwise progression to regulation (and away) can naturally not be observed in the case of implicit evolution because of the nature of this method. Instead, there the solutions evolve directly to regulated diauxic growth.

Anatomy of evolved solutions

In order to understand the adaptive pressures that act on the system, we analysed the populations that we obtained from implicit evolutionary simulations. To do this, we ran the evolution for at least 500 generations and then collected the entire population. From this we removed all unfit solutions — all solutions with fewer than 50 cells in total across all 32 environments. We collected populations in this way over a number of independent evolutionary runs. Altogether, this resulted in 1180 different parameter sets. Fig. 4 shows a boxplot characterising the distribution of parameters across the population. It compares evolutionary runs over two conditions: With limited space for porins and without limited porins.

In the case of a limitation of porins, a strong selective pressure seems to be limited to the leak expression of porins (leak1 and leak2), the parameter controlling the growth rate KG, dR and ukb. The latter two control the regulation of simulateneous growth. In particular, ukb is kept very low indicating that there is a strong selection pressure for diauxic growth in the system. The value of KG is a Hill constant
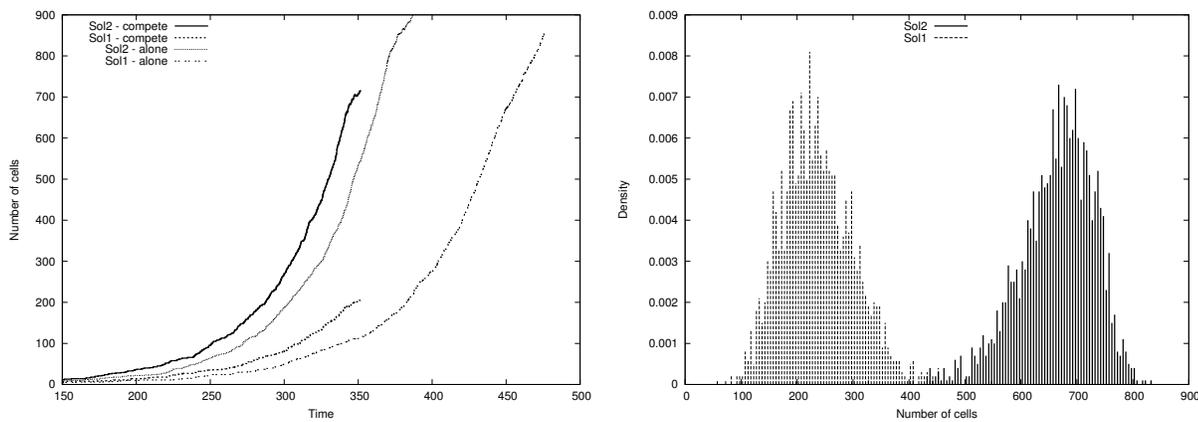
Figure 3: An example simulation comparing the solutions evolved at the first and second iteration (see Fig. 1). (**left**) The curves labelled "compete" show the result of a single simulation where the two solutions are competing for the same nutrient. Clearly, the second solution outcompetes the first one. The graph also shows a simulation of the first and second solution when simulated without competition (the curved labelled "alone"). In this case they grow similar cell numbers. (**right**) A histogram showing the distribution of cell numbers over 2000 competitive runs. There is hardly any overlap between the two distributions. The second solution clearly dominates the first.

and ensures that growth sets in at the correct availability of nutrient.

A notable result is the high leak rates. As can be seen from Table 1 the leak rate determines at what rate porins are expressed even in the absence of an activator. The activation mechanism used here is self-activation in the sense that porin is expressed in response to nutrient presence in the cell, but nutrient can only be present in the cell if there are porins. Hence, a small leak rate is required in order to kick-start the auto-activation of the porin expression system. Yet, a high leak rate essentially usurps the regulation of porin expression in the cell.

In real biological cells, there are two conceivable purposes for regulation of uptake mechanisms. The first is to control over-crowding of the intra-cellular space (or the cell surface for that matter) with proteins that are not needed. The second is to avoid wasting resources on producing proteins that are not needed. Neither is very relevant in the setup of this model for the following reason: our model has a direct negative feedback from surface crowding to porin expression. This means that there is no need for the cell to worry about unnecessary expression. At any time, the cell is either taking up N1 or N2; given the limited surface areas, this means that the negative feedback to expression is always fully engaged and only the porins that are required are actually expressed.

From this it appears that no (primary or direct) regulation of expression is required and therefore there are no disadvantages from a high leak rate. Yet, there are advantages. The main one is that the cell is able to respond quickly to change in circumstances. When N1 runs out the high leak rate of P2 makes it possible for production of porin 2 to start without delay. Hence, no lag phase evolves in these simulations.

The limitation of porins is the main rate limiting step in the growth dynamics of the cell. Hence, the solutions that are evolved without this limit show a fundamentally different pattern. When there is no limit on how many porins there can be on the cell surface, i.e. no limit on the maximum rate of importing nutrients, then the rate-limitation moves to the conversion rate of nutrient to cell energy, i.e. lk1 and lk2. Fig. 4 confirms that there is a strong selection pressure on these parameters. In nearly all members of the population they take the maximum allowed value.

On the other hand, the ukb and dR that were strongly selected in the case of limited surface area are now no longer under this pressure. When there is no limit on the surface area then it would be detrimental to block porin 2. Hence, the repression parameters need to be tuned so as to minimise the binding of R with P2. The precise values of the relevant parameters are not as important as the fact that the binding rate kb is low compared to the unbinding rate and that there is little R in the system, i.e. the value of dR must not be too low.

Relationship to real parameters

In our simulations, diauxic growth evolves robustly and reproducibly. However, there are crucial differences between the solutions that evolved and the real-world cells. Firstly, there is a small amount of simultaneous uptake of both nutrients. Secondly, there is no lag phase evolving. These two observations are connected but they are also significant with respect to the understanding of the evolution of diauxic growth.

The main parameter governing the length of the lag phase

Dominique Chu, David J. Barnes (2015) Evolving strategies for single-celled organisms in multi-nutrient environments. Proceedings of the European Conference on Artificial Life 2015, pp. 226-233
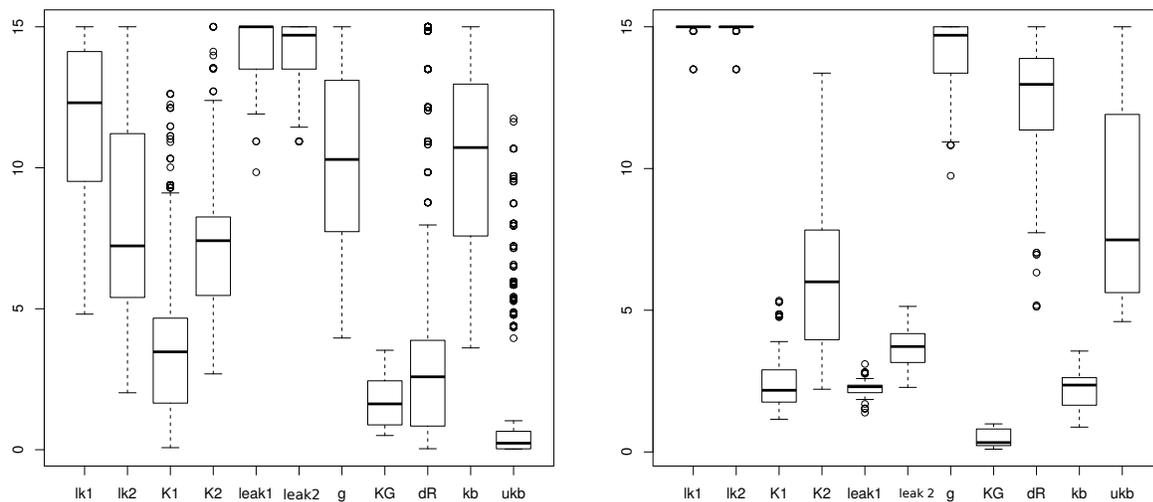
Figure 4: Comparing the parameters evolved. The left plot presents the parameters for the scenario where space on the surface is limited. The right plot indicates a scenario with unlimited space. The clear difference is the parameter `ukb` that evolved to very low values in the case of limited space.

(and the degree of simultaneous uptake) is `ukb`. In order to understand its influence better, we took an evolved solution and compared it with two variants. One where *ukb* was set to 15 — the maximum value permitted within our model — and one where it was set to zero. The former case corresponds to no diauxic growth at all and the latter to a very strong repression of `N2` uptake in the presence of the first nutrient. We confirmed in simulation that with this choice of parameters uptake is entirely sequential, i.e. no `N2` is used before `N1` is exhausted. This also leads to a clearly distinuishable lag phase in the growth curve. The latter case supressed diauxic growth completely and both nutrients are taken up simultaneously. Both of those variants are outperformed by the evolved solution (Fig. 5). This suggests that there is an optimal level of repression of the `P2` metabolism in our model.

Discussion

One possible objection to our simulations is that they are highly simplified with respect to real systems, and hence not comparable. It could be argued that any discrepancy between the evolved solutions and real cells is as irrelevant as any similarities would be coincidental.

Clearly, it is the case that the "bio-chemistry" used here is idealised with respect to the rather more complex web of interactions that regulate diauxic growth in real organisms. However, the complexity of the regulatory mechanisms is not the point. At the heart of the matter is the competitive nature of growing cells and the basic strategy. It is reasonable to assume that cells can adjust the length of the lag phase over evolutionary times. Sequential uptake of nutrients is a positively controlled evolved feature that cells could lose again. So, conceivably, they could also reduce the length of
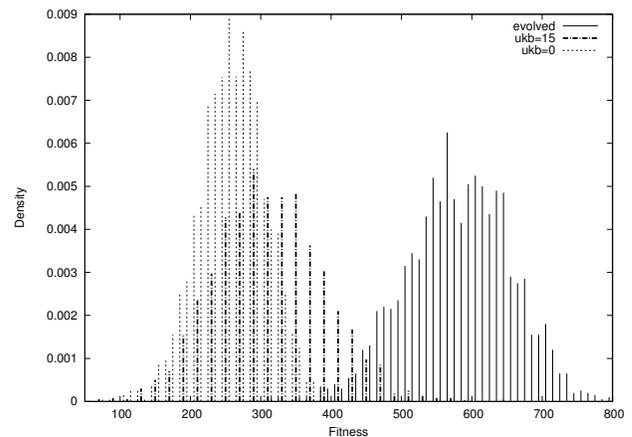

Figure 5: Fitness depends on the parameter `ukb`. The evolved solution shows a distribution of fitnesses. When the unbinding rate is set to zero or to very high values then the fitness goes down substantially compared to the evolved solution. This histograms were produced from 2000 simulations of the model.

the lag phase or even the strict sequential nature of nutrient uptake. This is confirmed by recent experimental findings that demonstrate this plasticity using artificial wet-lab evolution (Molenaar et al., 2009). For as long as the plasticity of dual nutrient uptake is established, then the details of the implementation are mostly irrelevant. This means that it makes sense to try to understand diauxic growth as a general strategy and it is not necessary to take into account the fine details of the bio-chemical implementation.

The question now is: Why is there an exclusive switch evolving in real cells but the model predicts a soft transition

between the two nutrients, with partially overlapping nutrient uptake? One could conjecture that there are delays in the activation of gene expression. Clearly, it takes some time to transcribe and translate genes. This delay in gene production is not represented in the model but, one could suspect, might be responsible for the lag phase during the switch between `N1` and `N2`.

While it is true that there is a minimum time for metabolic pathways to be activated, we think that this is at most a partial explantion for the lag phase. For one, the experimental evidence cited above suggests that lag phases are changeable, suggesting that there are some circumstances in which it is beneficial to have longer lag phases and others when it is better to have shorter ones. This implies that the fundamental limits on gene activation are not relevant. Yet, even in the absence of this evidence, the time required to switch on the genetic machinery does not matter for the question we ask. If a lag phase is detrimental to the fitness of the individual, then the cell could always avoid it by evolving simultaneous, or partially simultaneous utilisation of nutrients. The question remains: Why does it not do that?

In order to understand why lag phases are so common in bacteria, it is necessary to look for fitness trade-offs: something that compensates the cell for the lost growth in the sense that the total growth, compared to competitors, must be higher taking into account both growth and lag phases. This would include growth in different environments. For example, a typical *E.coli* cell will spend much time in environments that do not contain, say, lactose. In those environments, the cell will still express a certain amount of lactose porins by leak expression. This is wasted energy for as long as the cell does not actually get into lactose environments. Hence, if the cell never sees lactose, there will be an evolutionary pressure to adjust leak expression downwards.

On the other hand, if a cell is placed in a lactose environment, then it will be well served having a high leak rate. If leak expression of the lactose uptake system is efficiently repressed, then (on average) it will take a very long time to sense it and switch it back on. During this particular growth episode, this will lead to a long induction period. Or, if taken up in combination with glucose, there will be a long lag phase between glucose and lactose usage. If this scenario were true, a long lag phase would be a side effect of adaptation to a wide range of growth conditions.

Another hypothesis, not necessarily independent of the previous one, is that the lag phase is a consequence of computational cost. The switch from one nutrient to another can be considered as a computation (Zabet and Chu, 2010). Fast and accurate switching comes at energetic cost (Chu et al., 2011) in the sense that maintaining a gene regulatory network that can quickly change from one state to the next and does so accurately, comes at a metabolic cost. The cost of maintaining a highly-performing switch has to be traded off against the gain of fast switching during diauxic

growth. This scenario, if true, would entail that the computational/metabolic cost of sensing external nutrients fast enough is higher than the fitness gains from switching fast.

At present, this remains an open question. Addressing this remains a challenge for future work and requires integrative models of nutrient uptake taking into account the entire life-time of the cell. The significance of this question goes well beyond just uncovering a particular microbiological effects. Diauxic growth is fundamentally about cellular and molecular computing. In order to switch its state from metabolising one nutrient to metabolising another one, the cell has to sense external conditions and process this information. At the same time as it senses, it also has to ensure that the process of information processing is economical and properly counter-balanced by the gains of fast switching. As such, diauxic growth makes an interesting case study in biomolecular information processing that integrates questions from evolution, molecular computing and biology.

References

Boianelli, A., Bidossi, A., Gualdi, L., Mulas, L., Mocenni, C., Pozzi, G., Vicino, A., and Oggioni, M. (2012). A nonlinear deterministic model for regulation of diauxic lag on cellobiose by the pneumococcal multidomain transcriptional regulator celr. *PLoS One*, 7(10):e47393.

Boulineau, S., Tostevin, F., Kiviet, D., ten Wolde, P., Nghe, P., and Tans, S. (2013). Single-cell dynamics reveals sustained growth during diauxic shifts. *PLoS One*, 8(4):e61686.

Brückner, R. and Titgemeyer, F. (2002). Carbon catabolite repression in bacteria: choice of the carbon source and autoregulatory limitation of sugar utilization. *FEMS Microbiology Letters*, 209(2):141 – 148.

Chu, D., Zabet, N., and Hone, A. (2011). Optimal parameter settings for information processing in gene regulatory networks. *BioSystems*, 104:99–108.

Deutscher, J. (2008). The mechanisms of carbon catabolite repression in bacteria. *Current Opinion in Microbiology*, 11(2):87 – 93.

Gibson, M. and Bruck, J. (1998). An efficient algorithm for generating trajectories of stochastic gene regulation reactions. Technical Report CaltechPARADISE:1998.ETR026, California Institute of Technology.

Gillespie, D. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361.

Molenaar, D., van Berlo, R., de Ridder, D., and Teusink, B. (2009). Shifts in growth strategies reflect tradeoffs in cellular economics. *Molecular Systems Biology*, 5:323.

Monod, J. (1949). The growth of bacterial cultures. *Annual Review in Microbiology*, 3:371–349.

Stülke, J. and Hillen, W. (1999). Carbon catabolite repression in bacteria. *Current Opinion in Microbiology*, 2(2):195 – 201.

Zabet, N. and Chu, D. (2010). Computational limits to binary genes. *Journal of the Royal Society Interface*, 7(47):945–954.

Dominique Chu, David J. Barnes (2015) Evolving strategies for single-celled organisms in multi-nutrient environments. Proceedings of the European Conference on Artificial Life 2015, pp. 226-233